一种改进的核增量式更新算法

冯少荣，赖桃桃，张东站
FENG Shao-rong，LAI Tao-tao，ZHANG Dong-zhan

厦门大学 计算机科学系，福建 厦门 361005
Department of Computer Science，Xiamen University，Xiamen，Fujian 361005，China
E-mail：shaorong@xmu.edu.cn

FENG Shao-rong，LAI Tao-tao，ZHANG Dong-zhan.Improved incremental updating algorithm of computation of core. Computer Engineering and Applications，2010，46（20）：96-98.

Abstract： After analyzing the existence fast updating algorithm for computing an attributes core based on discernibility matrix——FUAC，this paper points out the reason that the FUAC has so high space complexity.Then an improved fast updating algorithm is proposed，which does not store discernibility matrix.Theoretical analysis shows that the improved incremental updating algorithm of the computation of a core has linear space complexity.

Key words： rough set；core；incremental updating

摘　要：深入分析基于差别矩阵的属性核快速更新算法——FUAC后，指出引起该算法空间复杂度高的原因，在此基础上提出了一种不存储差别矩阵的改进核增量式更新算法，主要考虑对象动态删除情况下核的更新问题。理论分析表明改进的核增量式更新算法有线性空间复杂度。

关键词：粗糙集；核；增量式更新
DOI：10.3778/j.issn.1002-8331.2010.20.027　**文章编号：**1002-8331（2010）20-0096-03　**文献标识码：**A　**中图分类号：**TP311

粗糙集理论是由波兰数学家Z.Pawlak1982年提出[1]，已经在机器学习、数据挖掘等领域中得到了较为广泛的应用。决策表信息系统是粗糙集理论的主要研究对象，决策表信息系统的约简是所有粗糙集理论和应用研究的焦点问题之一。决策表核属性的确定往往是信息约简的基础，因而探索研究求核的有效方法具有重要的实用价值。

目前有不少求解核的方法[2-7]，其中Hu提出的基于差别矩阵的求解核方法是经典的核求解方法之一[2]。Hu方法在某些情况下不能得到正确的核，叶东毅教授等人在Hu的差别矩阵定义基础上，提出新的差别矩阵并证明其求核方法是正确的[3]，但计算量大，为此文献[4]此提出了一种新的改进的差别矩阵及其求核方法，该方法在纠正Hu方法的错误的同时，可有效地降低计算代价，文献[7]利用基数排序的思想设计了一个基于正区域的快速求核算法。在文献[4]的基础上文献[5]提出基于差别矩阵的属性核快速更新算法，该算法具有比较高的空间复杂度。在文献[5-7]的基础上，提出了一种高效且易于实现的核增量式更新算法，较文献[5]的算法有效降低了空间复杂度，且与它一样有同样好的时间复杂度。

1　粗糙集理论的相关概念和理论

为了便于叙述，先对信息系统的属性约简以及属性核的概念进行简单介绍。

定义1　一个决策表信息系统$S=(U,A,V,f)$，其中，U为论域，$A=C\cup D$，$C\cap D=\Phi$，C为条件属性集，D为结果属性集，一般只含有一个属性，$V=\cup\{V_r|r\in A\}$是属性值的集合，V_r表示$r\in A$的值域，f：$U\times A\rightarrow V$是一个信息函数，它指定U中每个对象x的属性值。

定义2　给定一个决策表信息系统$S=(U,C\cup D,V,f)$，对于$X\subseteq U$和$R\subseteq C$，X的下近似集与X的上近似集分别定义为$R_*(X)=\cup\{Y|Y\in U/R 且 Y\subseteq X\}$和$R^*(X)=\cup\{Y|Y\in U/R 且 Y\cap X\neq\Phi\}$。

在信息系统（或决策表）中，若一些数据具有相同的条件属性而具有不同的分类，则称这类数据是不一致的；否则为一致的。

不失一般性，假设仅有一个决策属性D，其取值范围是1，2，…，k，由D导出的等价类构成划分：$\{\Psi_1,\Psi_2,\cdots,\Psi_k\}$，其中$\Psi_i=\{x\in U|f(x,D)=i\}$，$i=1,2,\cdots,k$。

定义3　设$P\subseteq C$，对划分$\{\Psi_1,\Psi_2,\cdots,\Psi_k\}$的$P$-近似精度为：

$$\gamma_P=\sum_{i=1}^{k}\frac{|P_*(\psi_i)|}{|U|}$$

定义4　设$P\subseteq C$，若$\gamma_P=\gamma_C$，且不存在$R\subset P$，$\gamma_R=\gamma_C$，则称P为C的一个（相对于决策属性D的）属性约简。所有C的属性约简的交称为C的核，记为$Core(C)$。

定义5　如果属性$a\in C$满足$\gamma_{C-\{a\}}<\gamma_C$，则称属性$a$为不可缺

基金项目：国家自然科学基金（the National Natural Science Foundation of China under Grant No.50604012）。
作者简介：冯少荣（1964-），男，博士，副教授，研究领域为数据库与数据挖掘；赖桃桃（1985-），男，硕士生，研究领域为数据挖掘；张东站，博士，副教授。
收稿日期：2009-04-09　**修回日期：**2009-06-04

少的，否则，称属性为冗余的。

性质1 属性 $a \in Core(C)$ 当且仅当 a 是不可缺少的属性。

2 已有的差别矩阵定义

为有效求核，Hu等学者提出一种简洁的利用改进差别矩阵来确定核的方法，但得出的结论在某些情况下是错误的，如该方法不能得到例1的核，却可得到例2的核，详见文献[3]。

例1 表1为二值数据表，其中共有5个元素和4个属性，$C=\{C_1,C_2,C_3\}$为条件属性集，D为决策属性。

在例1中删除第1个对象（即第1条记录）即得到表2。

表1 数据表（I）

元素	属性			
	C_1	C_2	C_3	D
x_1	1	0	1	1
x_2	1	0	1	0
x_3	0	0	1	1
x_4	0	0	1	0
x_5	1	1	1	1

表2 数据表（II）

元素	属性			
	C_1	C_2	C_3	D
x_1	1	0	1	0
x_2	0	0	1	1
x_3	0	0	1	0
x_4	1	1	1	1

针对Hu方法的缺陷，文献[3]提出了新的差别矩阵定义并给出求核方法，但计算量大。为改进文献[3]的不足，文献[4]提出了改进的差别矩阵定义以及求解核方法。文献[5]在文献[4]的基础上提出一种基于差别矩阵的属性核快速更新算法——FUAC。该算法在更新差别矩阵时仅需删除某一行及某一列，或插入某一行，因而可提高核的更新效率。

定义6[5] 对给定的信息系统IS，定义差别矩阵$\boldsymbol{M}1=\{m_{ij}\}$为：

$$m_{ij}=\begin{cases}\{a\in C{:}f(x_i,a)\neq f(x_j,a)\}, & \\ \qquad 当f(x_i,D)\neq f(x_j,D)且x_i\in U1,x_j\in U1 & \\ \{a\in C{:}f(x_i,a)\neq f(x_j,a)\},当x_i\in U1,x_j\in U2 & \\ \Phi, \qquad 其他 & \end{cases} \tag{1}$$

其中$U1=\bigcup_{i=1}^{k}C_*(\psi_i)$，$U2=U\text{-}U1$，$U2'=delrep(U2)$。函数$delrep(U2)$描述如下：

```
begin
  U2'=Φ;
  for 任意 x∈U2  do
    if不存在 y∈U2' 使得 ∀a∈C, f(x,a)=f(y,a)且f(x,D)≠f(y,D)
then{
      U2'=U2'∪{x};
        x.count=1;   //x.count表示与对象x不一致的对象个数
      } else {
        查找U2'中与x不一致的对象y;
        y.count=y.count+1;
      }
    return  U2';
  end.
```

定理1[5] 对于信息系统IS，若记$IDM(C,\boldsymbol{M}1)=\{m_{ij}|m_{ij}\in \boldsymbol{M}2$且$m_{ij}$为单个属性$\}$，则有$IDM(C,\boldsymbol{M}1)=Core(C)$，即当且仅当某个$m_{ij}$为单个属性时，该属性属于核$Core(C)$。

文献[5]分析得出FUAC算法的空间复杂度为$O(|U1|*(|U1|+|U2'|))$，可通过对差别矩阵中单个属性出现次数的更新来优化FUAC算法，类似于文献[6]，本文把优化后的FUAC称为OFUAC，其时间复杂度降为$O(|U1|+|U2'|)$。

3 新的求核方法及核增量式算法

为了节省存储空间，文献[5]把不一致对象$U2$约简为$U2'$。文中新算法不存储差别矩阵，所以不存在节省存储空间的问题，为了简化实现。用文献[4]的差别矩阵。

定义7[4] 对给定的信息系统IS，定义差别矩阵$\boldsymbol{M}2=\{m_{ij}\}$为：

$$m_{ij}=\begin{cases}\{a\in C{:}f(x_i,a)\neq f(x_j,a)\}, & \\ \qquad 当f(x_i,D)\neq f(x_j,D)且x_i\in U1,x_j\in U1 & \\ \{a\in C{:}f(x_i,a)\neq f(x_j,a)\},当x_i\in U1,x_j\in U2 & \\ \Phi, \qquad 其他 & \end{cases} \tag{2}$$

其中$U1=\bigcup_{i=1}^{k}C_*(\psi_i)$，$U2=U\text{-}U1$。

定理2[6] 对于信息系统IS，若记$DMSC(C)=\{(m,count):m\in\boldsymbol{M}2$且为单个属性，$count(\geqslant 1)$为$m$在$\boldsymbol{M}2$中出现的次数$\}$，$IDM(C,\boldsymbol{M}2)=\{m:(m,count)\in DMSC(C)\}$，其中，$\boldsymbol{M}2$为由定义7得到的差别矩阵，则有$IDM(C,\boldsymbol{M}2)=Core(C)$，即当且仅当某个$m$为单个属性时，该属性属于核$Core(C)$。

文献[7]提出了一种高效的静态求核算法，它求出核后直接输出核$Core(C)$。为了用于下文动态核更新算法，得把每个核属性及其个数存储于$DMSC(C)$。在文献[7]算法的基础上得到修改的求核算法如下：

算法1 求核算法

```
输入：U1、U2，其中U1、U2同定义6
输出：DMSC(C)
begin
  for each xi∈U1 do
    for each  xj∈U1∪U2  do {
      mij=Φ;
      flage=0;
      if ((xj∈U1 ,f(xi,D)!=f(xj,D) and i>j))|| xj∈U2 )
      for each  a∈C {
        flage++;
        if (f(xi,a)!=f(xj,a)) then mij=mij∪a;
        if (flage>1) break;
        }
      if (flage=1) then {
        if 存在(mij,count)∈DMSC(C) then
          从DMSC(C)删除(mij,count)后插入(mij,count+1);
          else 插入(mij,1)到DMSC(C)中;
        }
        }
    }
    end.
```

在对象删除情况下核的快速更新分以下两种情况进行：

（1）若$x\in U1$，则重新计算x与$U1$、$U2$中各对象间的属性核m_{ij}，并更新相应的$DMSC(C)$及$U1=U1\text{-}\{x\}$。

（2）若$x\in U2$，则找出$U2$中所有与x不一致的对象x_1，x_2，…，x_k。

重新计算x与$U1$中各对象间的属性核m_{ij}，并删除$DMSC(C)$中相应的m_{ij}及$U2=U2\text{-}\{x\}$。

若不一致对象个数c=2且与x不一致的对象为y，则$U1=U1+y$；$U2=U2\text{-}y$；重新计算y与$U1$、$U2$中各对象间的核属性m_{ij}，并将m_{ij}增加至$DMSC(C)$。

算法2 改进的优化快速核更新算法IOFUAC

输入：(1)$U1$、$U2$，由算法1计算得到的**DMSC(C)**

(2)删除对象为x

输出：$Core(C)$

begin

if $x \in U1$ then {

for each $x_i \in U1 \cup U2$ do { //把$DMSC(C)$中的与x相关的核属性删除

$m_{ij}=\Phi$；flage=0；

if $(((f(x_i,D)!=f(x,D)$ and $x_i \in U1)||x_j \in U2)$

for each $a \in C$ {

flage++；

if $(f(x_i,a)!=f(x,a))$ then $m_{ij}=m_{ij} \cup a$；

if (flage>1)break；

}

if (flage=1) then {

if 相应的$count$>1 then从$DMSC(C)$删除$(m_{ij},count)$后插入$(m_{ij},count-1)$；

else从$DMSC(C)$中删除$(m_{ij},count)$；

}

}

$U1=U1-\{x\}$；

}

else if $x \in U2$ then {

找出$U2$中所有与x不一致的对象$x_1,x_2,\cdots,x_k$；不一致对象个数记为k(含x)；

for each $x_i \in U1$ do {//把$DMSC(C)$中的与x相关的核属性删除

$m_{ij}=\Phi$；flage=0；

for each $a \in C$ {

flage++；

if $(f(x_i,a)!=f(x,a))$ then $m_{ij}=m_{ij} \cup a$；

if (flage>1) break；

}

if (flage=1) then {//k=2时，删除两个m_{ij}

if (k=2且相应的$count$>2)then从$DMSC(C)$删除$(m_{ij},count)$后插入$(m_{ij},count-2)$；

if (k>2且相应的$count$>1)then从$DMSC(C)$删除$(m_{ij},count)$后插入$(m_{ij},count-1)$；

else if(相应的$count$=1)从$DMSC(C)$中删除$(m_{ij},count)$；

}//end if (flage=1)

}//end for each $x_i \in U1$

$U2=U2-\{x\}$；

if不一致对象个数k=2 {

//若k=2，把上面找出$U2$中与x不相等的对象且标记为y；

$U1=U1+\{y\}$；$U2=U2-\{y\}$；

for each $x_i \in U1 \cup U2$ do {//把与y相关的核属性添加到$DMSC(C)$中

$m_{ij}=\Phi$；flage=0；

if $(((f(x_i,D)!=f(y,D)$ and $x_i \in U1)||x_i \in U2)$

for each $a \in C$ {

flage++；

if $(f(x_i,a)!=f(y,a))$ then $m_{ij}=m_{ij} \cup a$；

if (flage>1) break；

}

if (flage=1) then {

if 存在$(m_i,count) \in DMSC(C)$ then 从$DMSC(C)$删除$(m_{ij},count)$后插入$(m_{ij},count+1)$；

else 插入$(m_{ij},1)$到$DMSC(C)$中；

}

}//end for each $x_i \in U1 \cup U2$

}//end else if 不一致对象个数c=2

}//end else if $x \in U2$

由$DMSC(C)$得到核$Core(C)$；

end.

算法1分析[7]：

对于大型决策表，由于$|C| \leq |U|$，故其空间复杂度为$O(|U|)$。时间复杂度为$\max\{O(|C||U|),O(|C||U/C||U'_{pos}|)\}$。

算法2分析：

(1)若$x \in U1$，判断x的一致性的时间为$O(|U1|)$，把$DMSC(C)$中的与x相关的核属性删除的时间为$O(|U1|+|U2|)$，把x从$U1$删除时间最多为$O(U1)$，所以总的时间为$O(2*|U1|+2*|U2|)$。

(2)若$x \in U2$，判断x的一致性的时间为$O(|U1|)$，找出$U2$中所有与x不一致的对象$x_1,x_2,\cdots,x_k$的时间为$O(|U2|)$。

①不一致对象个数k>2，把$DMSC(C)$中的与x相关的核属性删除时间为$O(|U1|)$，把x从$U2$删除时间最多为$O(U2)$。

②当k=2时，除与①相同的时间外，把与y相关的核属性增加到$DMSC(C)$中时间为$O(|U1|+|U2|)$。

当$x \in U2$时，最多计算时间为$O(|U1|+|U2|+|U1|+|U2|+|U1|+|U2|)=O(3*|U1|+3*|U2|)$。

所以算法2的时间复杂度为$O(|U1|+|U2|)$，与优化快速核更新算法OFAUC的$O(|U1|+|U2'|)$相当。算法2空间复杂度为$O(|C|)$显著低于OFAUC的$O(|U1|*(|U1|+|U2'|))$；特别的在一致决策表中，OFAUC的时间复杂度为$O(|U|^2)$，此时$U1=U$。

4 实验

在内存为1 024 MB，CPU为PIV 2.9 GHz，操作系统为Windows XP的联想PC上，Eclipse下Java实现了文献[5]优化后的FUAC及文中的算法2。标记优化后的FUAC为OFUAC，算法2为IOFUAC。利用UCI上所提供的蘑菇数据库(mushroom)来进行实验，该数据库有8 124个对象。将下载的蘑菇数据库看作决策表，并进行以下两组实验：

(1)把8 124个对象全部作为基准决策表(基准决策表的含义是指该决策表生成的差别矩阵作为算法OFUAC和IOFUAC的输入)，然后从8 124个对象中依次选择200、500、800、1 124个对象作为增量，实验结果如图1所示。

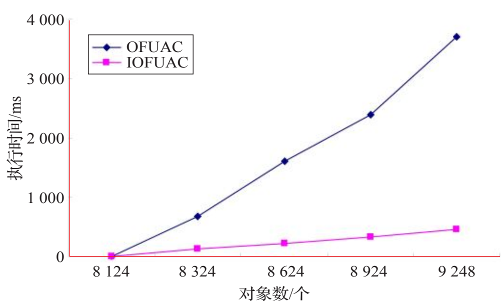

图1 第1组实验算法的执行时间

(2)由蘑菇数据库生成8 000个对象，其中不一致对象数1 000个，生成的8 000个对象作为基准决策表，从8 000个对象

(下转180页)

6 结论

对于模糊分类问题，给定一个模糊信息系统，不同的模糊条件属性对分类的贡献是不同的，有的贡献比较大，有的贡献比较小。从训练数据中计算模糊条件属性的模糊分类熵，并用它作为权值对相对模糊频率进行加权，这样将各个属性对分类的贡献通过加权和的形式融合起来，使比较重要的属性能起到比较大的作用，可综合考虑各个模糊条件属性对分类的贡献。基于这种基本思想，提出了基于属性模糊熵的模糊分类算法，实例分析和实验结果表明了这一算法的有效性。

参考文献：

[1] Zadeh L A.Fuzzy sets[J].Information and Control，1965，8：338-353.
[2] Nozaki K，Ishibuchi H，Tanaka H.Adaptive fuzzy rule-based classification system[J].IEEE Transactions on Fuzzy Systems，1996，4（3）：238-250.
[3] Simpson P K.Fuzzy min-max neural networks-part 1：Classification[J].IEEE Transactions on Neural Networks，1992，3（5）：776-786.
[4] Uebele F，Abe S，Lan M S.A neural network-based fuzzy classifier[J].IEEE Transactions on Systems，Man，and Cybernetics，1995，25（2）：353-361.
[5] Abe S，Thawonmas R.A fuzzy classifier with ellipsoidal regions[J].IEEE Transactions on Fuzzy Systems，1997，5（3）：358-368.
[6] Dubois D，Prade H，Testermale C.Weighted fuzzy pattern matching[J].Fuzzy Sets Systems，1988，28（3）：313-331.
[7] Grabisch M.Fuzzy integral in multicriteria decision-making[J].Fuzzy Sets and Systems，1995，69（3）：279-298.
[8] Chang R L P，Pavlidis T.Fuzzy decision tree algorithms[J].IEEE Transactions on Systems，Man，and Cybernetics，1977，7（1）：28-35.
[9] Umano M，Okamolo H，Hatono I，et al.Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis system[C]//Proceedings of Third IEEE International Conference on Fuzzy Systems，1994，3：2113-2118.
[10] Yuan Y F，Shaw M J.Induction of fuzzy decision trees[J].Fuzzy Sets System，1995，69（2）：125-139.
[11] 苗夺谦，李道国.粗糙集理论、算法与应用[M].北京：清华大学出版社，2008：34-66.
[12] Wang X Z，Yeung D S，Tsang E C C.A comparative study on heuristic algorithms for generating fuzzy decision trees[J].IEEE Transactions on Systems，Man，and Cybernetics，Part-B，2001，31（2）：215-226.
[13] Wang X Z，Zhai J H，Lu S X.Induction of multiple fuzzy decision trees based on rough set technique[J].Information Sciences，2008，178（16）：3188-3202.
[14] Blake C L，Merz C J.UCI repository of machine learning databases[EB/OL].（1996）.http：//www.ics.uci.edu/~mlearn/MLRepository.html.
[15] Chen S M，Shie J D.Fuzzy classification systems based on fuzzy information gain measures[J].Expert Systems with Applications，2009，36（3）：4517-4522.

（上接98页）

中依次选择200、500、800、1 000个对象作为增量，实验结果如图2。

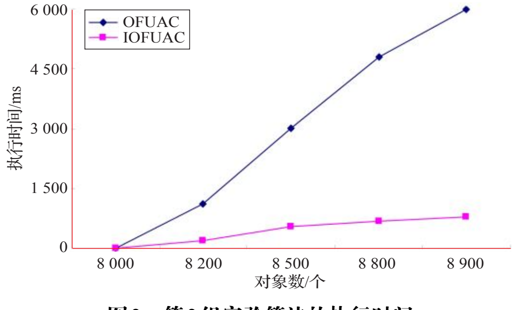

图2 第2组实验算法的执行时间

实验分析：

时间复杂度分析显示，算法OFAUC与IOFAUC都为线性的。不过OFAUC把核属性存储于差别矩阵，所以要修改及遍历差别矩阵相应的行和列。而IOFAUC少了遍历差别矩阵的时间且优化了核属性计算算法，当两个对象的属性差异度大于1时，停止此次计算，即程序中的if（flage>1）break；所以它的计算时间比较少。

实验监测显示，因OFUAC存储差别矩阵，每次实验至少增加内存200 MB；而IOFUAC一直保持稳定且较小的内存增加，为20 MB左右。所以IOFUAC较OFUAC的另一重要优势为IOFUAC可以应对大数据集的挑战。

5 结论

提出一种改进的属性核快速更新算法，主要考虑对象动态删除情况下核的更新问题。提出的算法与文献[5]的算法一样具有线性时间复杂度且计算量少于文献[5]算法。更重要的是本文算法有线性空间复杂度；而文献[5]算法有较高的空间复杂度，特别在一致性决策表中其空间复杂度为平方级。所以本文算法可以有效地应对大数据集的挑战。

参考文献：

[1] Pawlak Z.Rough sets[J].International Journal of Information and Computer Science，1982，11（5）：341-356.
[2] Hu X H，Cercone N.Learning in relational databases：A rough set approach[J].Computational Intelligence：An International Journal，1995，11（2）：323-338.
[3] 叶东毅，陈昭炯.一个新的差别矩阵及其求核方法[J].电子学报，2002，30（7）：1086-1088.
[4] 杨明，孙志挥.改进的差别矩阵及其求核方法[J].复旦学报：自然科学版，2004，43（5）：865-868.
[5] 杨明，杨萍.基于差别矩阵的属性核快速更新算法[J].控制与决策，2007，22（4）：453-456.
[6] 杨明.一种基于改进差别矩阵的核增量式更新算法[J].计算机学报，2006，29（3）：407-413.
[7] 徐章艳，杨炳儒，蔡卫东，等.一个基于正区域的快速求核算法[J].系统工程与电子技术，2006，28（12）：1902-1904.